AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Original) A method of processing data comprising:

processing a function using a processor operable to perform a plurality of functions, said processor having interrupts enabled;

receiving an interrupt at said processor;

suspending processing of said function;

accessing at least one control parameter, said at least one control parameter indicating whether processing of said function should be resumed from a point where it was interrupted or whether said function should be repeated following said interrupt;

following completion of said interrupt continuing processing of said function either at a start of said function or at a point at which it was interrupted dependent upon said control parameter.

2. (Original) A method according to claim 1, wherein said function processed by said processor may comprise an application, a system software routine, a thread, or multiple processing steps defined by software.

3. (Original) A method according to claim 1, comprising a further step of:

controlling said processor to store a restart address at which said processor should continue processing in dependence upon said at least one control parameter.

4. (Original) A method according to claim 3, wherein if said at least one control parameter indicates that said function is to be repeated following said interrupt said method comprises an additional step of:

accessing a further control parameter, said further control parameter being indicative of whether said function has idempotence or not; and

following completion of said interrupt continuing processing of said function at a start of said function if said further control parameter indicates said function to have idempotence or at a fix-up routine to be performed before said function is restarted if said control parameter indicates said function not to have idempotence.

5. (Original) A method according to claim 4, wherein said step of controlling said processor to store a restart address at which said processor should continue processing is dependent upon both said at least one control parameter and said further control parameter, said restart address being said start of said function if said control parameter indicates that said function is to be repeated and said further control parameter indicates said function to have idempotence or an address of said fix-up routine to be performed before said function is restarted if said further control parameter indicates said function not to have idempotence.

6. (Original) A method according to claim 1, comprising a further step of:

controlling said processor to retrieve stored data relating to a restart address at which said processor should continue processing in dependence upon at least one control parameter;

following completion of said interrupt, continuing processing of said function from the stored restart address.

7. (Original) A method according to claim 6, wherein said processor is operable in a plurality of modes, said method comprising the additional steps of:

prior to initiation of processing of said function, switching said processor to a mode in which interrupts are automatically disabled on entry to said mode;

storing an address at which said processor switched mode; and

on initiating said function, said function controlling said processor to enable interrupts.

8. (Original) A method according to claim 7, wherein

said at least one control parameter indicates that said restart address comprises said address at which the processor switched mode.

9. (Currently Amended) A method according to claim 4 ~~and 7~~, wherein

said further control parameter indicates a fix-up routine should be processed prior to restarting said function, said fix-up routine being operable to restore a state of said processor such that said function can be restarted and have idempotence and to disable interrupts during its processing.

10. (Original) A method according to claim 9, wherein on receipt of said interrupt said processor processes said fix-up routine prior to handling said interrupt.

11. (Original) A method according to claim 9, wherein following completion of said interrupt, said processor goes to an address at which the processor switched modes, and following switching modes, said processor is operable to process said fix-up routine logic prior to restarting said function.

12. (Original) A method according to claim 7, wherein said control parameter comprises a mode identifier.

13. (Original) A method according to claim 7, wherein said mode is a monitor mode, said processor being operable in a plurality of domains comprising a secure domain and a non-secure domain, such that when said processor is executing a program in a secure domain said program has access to secure data which is not accessible when

said processor is operating in a non-secure domain, switching between the domains only being possible when said processor is operating in monitor mode.

14. (Original) A method according to claim 1, comprising the additional step of:

following continuing of processing of said function, disabling interrupts.

15. (Original) A method according to claim 1, wherein if said at least one control parameter indicates that said function should be resumed at a point that it was interrupted following processing of said interrupt, said processor stores an address at which the function was interrupted as a restart address in an exception register, while if said control parameter indicates that said function should be restarted at a start of said function following processing of said interrupt, the exception register is not updated following said interrupt.

16. (Original) An apparatus for processing data, said apparatus comprising:

a processor operable to perform a plurality of functions and comprising:

a control parameter storage element operable to store a control parameter indicative of whether processing of a function should be resumed from a point where it was interrupted or whether it should be repeated following said interrupt;

an interrupt signal input port;

an interrupt enable/disable selector; and

function logic operable to control said processor to perform a function; wherein said processor is operable to process function logic and in response to receipt of an interrupt signal when said interrupt selector is enabled, to suspend processing of said function logic, and dependent on a value of said control parameter stored in said control parameter storage element to continue processing of said function either at a start of said function or at a point at which it was interrupted following completion of said interrupt.

17. (Original) An apparatus according to claim 16, wherein said function logic may comprise an application, a system software routine, a thread, or multiple processing steps defined by software.

18. (Original) An apparatus for processing data according to claim 16, said apparatus further comprising a restart address storage element operable to store a restart address at which said processor should continue processing following an interrupt, said processor being operable to store a restart address in said restart address storage element in dependence upon said at least one control parameter.

19. (Original) An apparatus for processing data according to claim 18, said processor comprising a further control parameter storage element, said further control parameter indicating whether said function has idempotence or not and fix-up routine logic;

said processor being operable to access said further control parameter if said control parameter indicates that said function is to be repeated following said interrupt, and following completion of processing of said interrupt said processor is operable to continue processing of said function at a start of said function if said further control parameter indicates said function to have idempotence or at a fix-up routine to be performed before said function is restarted if said control parameter indicates said function not to have idempotence.

20. (Original) An apparatus for processing data according to claim 19, wherein said processor is operable to store a restart address in said restart address storage element in dependence upon both said at least one control parameter and said further control parameter, said restart address being a start of said function if said control parameter indicates the function is to be repeated and said further control parameter indicates said function to have idempotence or an address of a fix-up routine to be performed before said function is restarted if said further control parameter indicates said function not to have idempotence.

21. (Original) An apparatus for processing data according claim 16, wherein said processor is operable to retrieve stored data relating to a restart address at which said processor should continue processing following an interrupt in dependence upon at least one control parameter.

22. (Original) An apparatus for processing data according to claim 21, said processor being operable in a plurality of modes, including a mode in which interrupts are automatically disabled upon entry to said mode, said function logic being executable in this mode and comprising logic to enable interrupts, said processor being operable to store an address at which it switched modes.

23. (Original) An apparatus for processing data according to claim 22 wherein said control parameter indicates that said restart address comprises said address at which said processor switched mode.

24. (Currently Amended) An apparatus for processing data according to claim 22 ~~and claim 19~~, said apparatus further comprising fix-up routine logic, and wherein said processor is operable to process said fix-up routine logic prior to restarting said function in dependence upon said further control parameter.

25. (Original) An apparatus for processing data according to claim 22, wherein said processor is operable to process said fix-up routine prior to handling said interrupt.

26. (Original) An apparatus for processing data according to claim 22, said processor is operable following completion of processing said interrupt to go to an

address at which it switched modes, and following switching modes, the processor is operable to process said fix-up routine logic prior to restarting said function.

27. (Original) An apparatus for processing data according to claim 20, wherein said control parameter comprises said mode identifier.

28. (Original) An apparatus for processing data according to claim 20, wherein said processor is operable in a plurality of domains comprising a secure domain or a non-secure domain, such that when said processor is executing a program in said secure domain the program has access to secure data which is not accessible when the processor is operating in said non-secure domain, switching between said domains only being possible when said processor is operating in a monitor mode.

29. (Original) An apparatus for processing data according claim 16, wherein said function logic comprises logic to disable interrupts.

30. (Original) An apparatus for processing data according to claim 17, the processor comprising interrupt exception registers, said restart address storage element comprising an interrupt exception register.

31. (Original) A computer program product comprising:

function logic operable to control a processor to perform a function: and

disable interrupt logic operable to control said processor to disable interrupts; wherein

a first portion of said function logic operable to control said processor to perform a first portion of said function, has idempotence such that it does not alter a state of any part of the processor which would affect the repeatability of the function and is operable to be executed by said processor before said disable interrupt logic and a final portion of said function logic operable to control said processor to complete said function is operable to be executed after said disable interrupt logic is executed.

32. (Original) A computer program product according to claim 31, further comprising control parameter logic operable to control said processor to write a control parameter, said control parameter indicating that following an interrupt said function should be restarted, said control parameter logic being operable to be executed before said first portion of the function logic.

33. (Original) A computer program product according to claim 32, further comprising:

a further portion of said function logic and further control parameter logic; wherein

said further portion of said function logic is operable to be executed after said first portion of said function logic and before said interrupt disable logic, said further portion of said function logic not having idempotence; and

said further control parameter logic is operable to control said processor to write a further control parameter, said further control parameter indicating that following an interrupt a fix-up routine should be run prior to restarting said function.

34. (Originalf) A computer program product according to claim 33, further comprising fix-up routine logic, said fix-up routine being operable to be processed following indication by said further control parameter and on processing said fix-up routine is operable to restore a state of said processor such that said function logic can be restarted and have idempotence and wherein said fix-up routine is operable to disable interrupts during its processing.

35. (Original) A computer program product according to claim 31, said computer program product further comprising interrupt enable logic, said interrupt enable logic being operable to be performed before said function logic.

36. (Original) A computer program product according to claim 31, wherein:

said first portion of said function logic comprises logic to control said processor to store data to a stack but not to update a stack pointer, and said final portion of said

function logic operable after said disable interrupt logic comprises logic operable to update a stack pointer with respect to data stored to said stack during execution of said first portion of the function logic.